



## Prediksi *Not Operational Transaction* Menggunakan *Logistic Regression* pada Bank XYZ di Kota Kupang

Radius Tanone<sup>1)</sup>, Arnold B Emmanuel<sup>1)</sup>

<sup>1)</sup>Fakultas Teknologi Informasi Universitas Kristen Satya Wacana  
Jl. Diponegoro No. 52-60, Salatiga, 50711  
Email : radius.tanone@uksw.edu

Received: 10-05-2020

Riwayat artikel:  
Revised: 31-08-2020

Accepted: 31-08-2020

### Abstract

*Bank XYZ is one of the banks in Kupang City, East Nusa Tenggara Province which has several ATM machines and is placed in several merchant locations. The existing ATM machine is one of the goals of customers and non-customers in conducting transactions at the ATM machine. The placement of the ATM machines sometimes makes the machine not used optimally by the customer to transact, causing the disposal of machine resources and a condition called Not Operational Transaction (NOP). With the data consisting of several independent variables with numeric types, it is necessary to know how the classification of the dependent variable is NOP. Machine learning approach with Logistic Regression method is the solution in doing this classification. Some research steps are carried out by collecting data, analyzing using machine learning using python programming and writing reports. The results obtained with this machine learning approach is the resulting accuracy value of 0.507 for its classification. This means that in the future XYZ Bank can classify NOP conditions based on the behavior of customers or non-customers in making transactions using Bank XYZ ATM machines.*

**Keywords:** Machine Learning, Logistic Regression, Python, Classification

### Abstrak

Bank XYZ merupakan salah satu bank di Kota Kupang, Propinsi Nusa Tenggara Timur yang memiliki beberapa mesin ATM dan ditempatkan di beberapa lokasi *merchant*. Mesin ATM yang ada menjadi salah satu tujuan nasabah maupun non-nasabah dalam melakukan transaksi di mesin ATM tersebut. Penempatan mesin-mesin ATM tersebut terkadang membuat mesin tidak dipakai secara maksimal oleh nasabah untuk bertransaksi sehingga menyebabkan pembuangan sumberdaya mesin dan terjadi kondisi yang disebut *Not Operational Transaction* (NOP). Dengan adanya data yang terdiri dari beberapa variabel independen dengan tipe numerik maka perlu diketahui bagaimana klasifikasi terhadap variabel dependen yaitu NOP. Pendekatan *machine learning* dengan metode *Logistic Regression* menjadi solusi dalam melakukan klasifikasi ini. Beberapa langkah penelitian dilakukan dengan pengumpulan data, analisa menggunakan machine learning menggunakan pemrograman python dan penulisan laporan. Hasil yang diperoleh dengan pendekatan *machine learning* ini adalah dihasilkannya nilai akurasi sebesar 0,507 untuk klasifikasinya. Hal ini berarti ke depan Bank XYZ dapat melakukan klasifikasi kondisi

NOP berdasarkan perilaku nasabah atau non-nasabah dalam bertransaksi menggunakan mesin ATM Bank XYZ.

**Kata kunci:** *Machine Learning, Logistic Regression, Python, Klasifikasi*

## Pendahuluan

Gubernur Bank Indonesia (BI) Agus DW Martowardojo menyatakan, himbauan Presiden untuk melakukan integrasi mesin ATM sejalan dengan langkah BI menerapkan Gerbang Pembayaran Nasional (National Payment Gateway/NPG). Hal ini merupakan tanggapan dari beliau selaku Gubernur BI terhadap perintah presiden Jokowi untuk mengurangi jumlah mesin ATM [1]. Bank XYZ menempatkan beberapa mesin ATM yang bisa disebut Merchant di beberapa pusat belanja di Kota Kupang, Propinsi Nusa Tenggara Timur. Hal ini untuk membantu nasabah dan atau *non*-nasabah Bank untuk bertransaksi secara *non*-tunai di tempat-tempat khusus perbelanjaan. Dari data transaksi *non*-tunai yang diperoleh Bank, setiap tahun beberapa kategori data dilihat untuk dianalisis dalam pengambilan kebijakan perlu tidaknya penambahan mesin ATM di posisi tertentu untuk membantu nasabah dalam bertransaksi. Beberapa kategori data yang dipakai diantaranya *Not Operational Transaction* (NOP), besar dana pertransaksi, *Merchant Discount Rate* (MDR ON US), MDR OFF US, NOP, Alamat, Tanggal Transaksi dan lain sebagainya. MDR sendiri merupakan tarif yang dibebankan kepada pedagang untuk layanan pemrosesan pembayaran pada transaksi kartu debit dan kredit [2].

Melihat banyaknya kategori data yang dipakai oleh Bank, perlu adanya analisa secara mendalam untuk melihat perilaku masyarakat dalam hal ini nasabah dan atau *non*-nasabah dalam bertransaksi menggunakan mesin ATM Bank tersebut. Apabila data yang diolah lebih dari satu tahun maka tentunya akan ada begitu banyak data yang sulit juga untuk diolah ke depannya. Berdasarkan wawancara dengan pihak Bank XYZ, *Not Operational Transaction* (NOP) merupakan masalah yang sering dialami oleh pihak penyedia Mesin (Perbankan), oleh karena semakin banyak tempat-tempat usaha yang ingin menggunakan mesin nontunai (EDC) tersebut namun ketersediaan mesin EDC yang minim oleh karena banyak pengguna atau pemilik usaha ingin memasang mesin namun tidak melakukan transaksi selama 30 hari. Sehingga kendala-kendala yang dialami adalah mesin EDC tidak bertransaksi selama 1 bulan (30 hari). Selain itu kegagalan dalam bertransaksi karena jaringan dana tidak masuk ke rekening *Merchant* (pemilik usaha). Masalah lain yang dihadapi juga adalah kurangnya *maintenance* dari pihak pemilik usaha ke kasir-kasir. *Merchant* (pemilik usaha) tidak ingin menanggung biaya MDR.

Saat ini Bank XYZ menyediakan lebih dari 30 mesin ATM *non*-tunai yang ditempatkan di beberapa gerai *merchant* di Kota Kupang. Dari berbagai mesin yang sudah dipasang, ada beberapa mesin yang tidak dipakai secara maksimal sehingga menyebabkan pemborosan sumberdaya mesin. Mesin yang tidak beroperasi secara maksimal ini menyebabkan kondisi yang dinamakan dengan NOP. Namun melihat banyak faktor yang dipakai dalam menentukan NOP atau tidak NOP sebuah mesin ATM maka diperlukan analisis mendalam. Untuk menyelesaikan masalah ini pihak Bank dapat melakukan prediksi secara kategori, mesin mana yang termasuk NOP dan mana yang tidak NOP dari faktor-faktor yang terjadi di lapangan.

Regresi logistik digunakan secara luas untuk menguji dan menggambarkan hubungan antara variabel respons biner (misalnya, 'Sukses' atau 'kegagalan') dan satu set variabel predictor [3]. Melihat data nasabah (dalam bentuk kategori) pada penggunaan ATM di Bank XYZ maka analisis menggunakan Regresi Logistik menjadi tepat guna melakukan prediktif berupa NOP atau tidaknya transaksi dari Bank tersebut. Dengan demikian maka perlu tambahan *tools* yang mampu membantu proses analisa menggunakan Regresi Logistik ini seperti pemrograman python. Penerapan *Logistic Regression* dengan python merupakan salah satu pendekatan *Machine Learning* yang dipakai dalam menganalisa data yang ada di Bank XYZ ini. Penelitian ini menggunakan metode Regresi Logistik dan Python sebagai salah satu pendekatan *Machine Learning* untuk membantu pihak Bank dalam melakukan analisa terkait prediksi NOP diantara beberapa variabel tersebut sehingga bisa dipakai untuk pengambilan keputusan pihak Bank ke depannya.

Sehingga yang menjadi rumusan masalah dalam penelitian ini adalah bagaimana membantu pihak Bank XYZ dalam melakukan prediksi terjadinya NOP dari banyak variabel transaksi yang terjadi di mesin-mesin ATM menggunakan pendekatan *Machine Learning* dengan metode *Logistic Linear*. Adapun tujuan dari penelitian ini yaitu membantu pihak Bank dalam menemukan pola hubungan data transaksi perilaku nasabah dan atau *non*-nasabah di mesin ATM Bank yang nantinya dapat diketahui apakah dalam kondisi NOP atau tidak NOP. Manfaat dari penelitian ini adalah membantu pihak Bank dalam mendapatkan analisa yang mendalam guna menghasilkan keputusan ke depannya yang lebih baik dalam hal kebijakan menempatkan mesin ATM Bank XYZ di *merchant* yang tepat berdasarkan jenis transaksi yang dilakukan oleh nasabah dan *non*-nasabah Bank XYZ. Adapun batasan masalah dalam penelitian ini yaitu data yang digunakan dalam kurun waktu antara dua tahun saja.

## Kajian Pustaka

***Machine Learning*** adalah metode yang digunakan untuk membuat program yang bisa belajar dari data. Berbeda dengan program komputer biasa yang statis,

program *machine learning* adalah program yang dirancang untuk mampu belajar sendiri. Cara belajar program *machine learning* mengikuti cara belajar manusia, yakni belajar dari contoh-contoh. *Machine learning* akan mempelajari pola dari contoh-contoh yang dianalisa, untuk menentukan jawaban dari pertanyaan-pertanyaan berikutnya [4]. Ada tiga jenis *Machine Learning* diantaranya *Supervised Learning*, *Unsupervised Learning* dan *Reinforcement Learning*. Dalam penelitian ini digunakan jenis *supervised learning* dengan pendekatan *Logistic Regression*.

***Logistic Regression*** adalah sebuah pendekatan untuk membuat model prediksi seperti halnya regresi linear atau yang biasa disebut dengan istilah *Ordinary Least Squares (OLS) regression*. Perbedaannya adalah pada regresi logistik, peneliti memprediksi variabel terikat yang berskala dikotomi. Skala dikotomi yang dimaksud adalah skala data nominal dengan dua kategori, misalnya: Ya dan Tidak, Baik dan Buruk atau Tinggi dan Rendah [5].

**Python** adalah bahasa pemrograman lintas platform dan karenanya pengembang dapat menggunakan Python pada PC Windows, Linux Box atau Apple Mac dll. Ini berarti tidak terikat untuk jenis sistem operasi tertentu [6]. Python adalah bahasa pemrograman yang *open source* membuatnya dapat digunakan dan didistribusikan secara bebas, bahkan untuk penggunaan komersial. Selain itu, python juga *user-friendly* dan mudah untuk dipelajari. Python dapat dengan mudah dikenal lebih jauh apakah anda seorang *programmer* pertama kali atau telah berpengalaman dengan bahasa pemrograman lain [7]. Dalam penelitian ini, bahasa pemrograman python digunakan sebagai alat bantu dalam memproses regresi logistik untuk melakukan prediksi NOP yang terjadi.

Untuk mendukung penelitian ini, digunakan beberapa penelitian terdahulu sebagai acuan ilmiah. Adapun penelitian pertama yang dilakukan oleh Le.H.H dan Viviani. J-L dengan judul “*Predicting bank failure: An improvement by implementing a machine-learning approach to classical financial ratios*” pada tahun 2018. Penelitian ini membandingkan akurasi dua pendekatan: teknik statistik tradisional dan teknik pembelajaran mesin, yang berupaya untuk memprediksi kegagalan bank. Sampel 3000 bank di Amerika Serikat (1438 kegagalan dan 1562 bank aktif) diselidiki oleh dua pendekatan statistik tradisional (Analisis

diskriminatif dan regresi logistik) dan tiga pendekatan pembelajaran mesin (jaringan saraf tiruan, Mesin *Vector Support*, dan k-tetangga terdekat). Dengan menggunakan data 5 tahun terakhir sebelum data tersebut menjadi tidak terpakai, Hasil empiris mengungkapkan bahwa jaringan saraf tiruan dan metode tetangga terdekat k adalah yang paling akurat [6].

Penelitian lain yang dilakukan oleh Ginting W.A dan kawan-kawan dengan judul “Faktor-Faktor Pelaporan Keuangan Berbasis *Website* Dengan Model Regresi Logistik” pada tahun 2019. Studi ini menunjukkan bukti empiris apakah ada pengaruh pada ukuran perusahaan, profitabilitas, dan reputasi auditor pada pengungkapan pelaporan keuangan berdasarkan pada situs web dan juga yang tidak berdasarkan situs web perusahaan pada tahun 2016. Dari penelitian yang dilakukan, diperoleh hasil bahwa peluang bagi perusahaan *non-finansial* yang menerapkan *Internet Financial Report* (IFR) lebih besar daripada perusahaan yang tidak menerapkan IFR. Alasan ini mendukung bahwa keberadaan internet melalui IFR telah banyak digunakan untuk memperluas jaringan bisnis di setiap entitas bisnis melalui situs web perusahaan dibandingkan dengan perusahaan yang belum mengimplementasikannya [7].

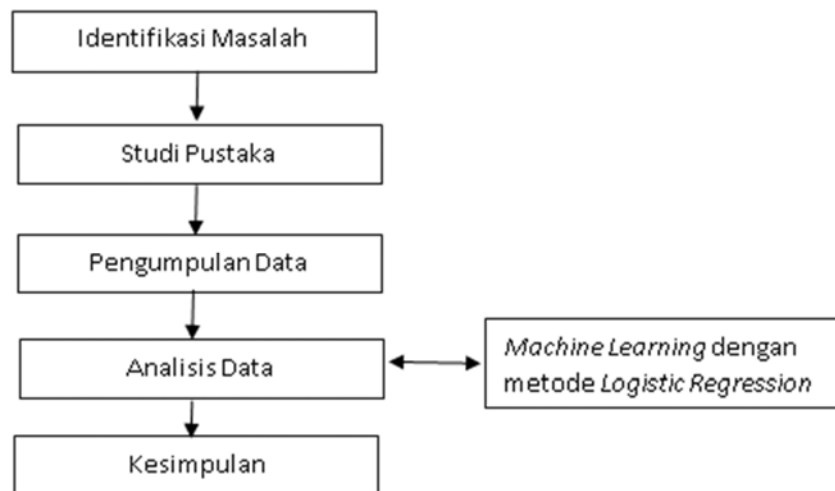
Dari penelitian terdahulu yang sudah dilakukan, kesamaan dengan penelitian ini adalah penggunaan *machine learning* sebagai pendekatan dalam menganalisis data. Sedangkan metode yang dipakai dalam *machine learning* adalah metode *logistic regression*. Untuk studi kasus tentunya berberda dimana data yang akan diambil pada Bank XYZ adalah data dalam kurang lebih tiga tahun terakhir. Selain itu, dengan metode *logistic regression*, dilakukan analisa terhadap data kategori untuk melakukan prediksi pada NOT yang terjadi di Bank XYZ.

### **Metode Penelitian**

Penelitian ini menggunakan metode kuantitatif karena disesuaikan dengan jenis data dan metode pengolahan data. Penelitian kuantitatif adalah suatu proses menemukan pengetahuan yang menggunakan data berupa angka sebagai alat menganalisis keterangan mengenai apa yang ingin diketahui [8]. Sedangkan menurut Sugiyono, metode penelitian kuantitatif dapat diartikan sebagai metode penelitian yang berlandaskan pada filsafat positivisme, digunakan untuk meneliti pada populasi atau sampel tertentu. Teknik pengambilan sampel pada umumnya dilakukan secara random, pengumpulan data menggunakan instrumen penelitian, analisis data bersifat kuantitatif/statistik dengan tujuan untuk menguji hipotesis yang telah ditetapkan [9].

Pendekatan pada penelitian ini adalah deskriptif yang mana penelitian deskriptif menurut Sugiyono [9] adalah "Metode penelitian deskriptif ini dilakukan untuk mengetahui keberadaan variable mandiri, baik hanya pada sat variabel atau

lebih (variabel yang berdiri sendiri atau variabel bebas) tanpa membuat perbandingan variabel itu sendiri dan mencari hubungan dengan variabel lain.” Sedangkan populasi dari penelitian ini adalah Bank XYZ di NTT sedangkan sampel penelitian adalah Bank XYZ di Kota Kupang. Adapun tahapan penelitian yang dilakukan secara yaitu identifikasi masalah, studi pustaka, pengumpulan data, analisis dan penulisan hasil kesimpulan. Tahapan penelitian dapat dilihat pada Gambar 1 di bawah ini.

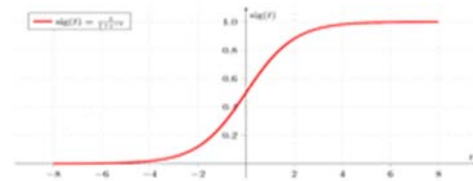


**Gambar 1.** Tahapan Penelitian

Adapun Tahapan yang akan dilakukan dalam penelitian ini diantaranya identifikasi masalah dimana pada tahapan ini dilakukan penelusuran terhadap tempat penelitian untuk melihat sumber permasalahan yang akan diteliti. Selain itu juga ditentukan penggambaran deskriptif terhadap masalah yang didapatkan sampai pada tahapan akhir dari penelitian ini. Selain itu pada tahapan ini juga ditentukan populasi dan sample juga alat ukur, agar dapat dipakai dalam proses pengambilan data, analisis dan penulisan hasil kesimpulannya. Selanjutnya tahapan Studi Pustaka dimana pada tahapan ini dilakukan telaah terhadap sumber pustaka agar dapat mendukung penelitian ini secara utuh. Telaah pustaka dapat dimulai dari mengkaji penelitian terdahulu serta melihat teori-teori yang akan dipakai untuk mendukung penelitian ini.

Langkah selanjutnya adalah pengumpulan data dimana pada tahapan ini ditentukan teknik pengumpulan data melalui wawancara kepada pihak Bank XYZ dan pengumpulan data sekunder seperti dokumen yang dapat dipakai untuk mendukung penelitian ini. Berikutnya adalah tahapan analisis dimana pada tahapan ini dilakukan analisis menggunakan *Logistic Regression* untuk melihat gambaran deskriptif yang ada di Bank XYZ. Analisis ini menggunakan *logistic function* dan

pemrograman python dengan pendekatan *Machine Learning* tipe *supervised learning*. Kurva fungsi logistic dapat dilihat pada Gambar 2.



*Logistic Function*

**Gambar 2.** Bentuk *Logistic Function*

Tahapan terakhir yang dilakukan adalah penulisan hasil dan kesimpulan penelitian. Tahapan ini berarti merupakan tahapan akhir dari penelitian dimana hasil dari penelitian ini selanjutnya akan ditulis dan kemudian dipublikasikan.

## Hasil dan Pembahasan

Dalam menerapkan *Logistic Regression* sebagai salah satu pendekatan *Machine Learning* maka adapun hasil yang diperoleh setelah melalui beberapa tahapan menggunakan pemrograman python. Hasil yang diperoleh akan dijelaskan satu persatu mulai dari langkah awal hingga menemukan hasil probabilitas NOP yang terjadi pada Bank XYZ. Awalnya data dalam bentuk CSV disiapkan dimana data dalam bentuk kategorial bertipe integer. Setelah itu diperlukan *code* python untuk mengolah hasil tersebut. Berikut pembahasannya:

### Kode Program 1. *Import Library* pada python

```
1. import pandas as pd
2. import pylab as pl
3. import numpy as np
4. import scipy.optimize as opt
5. from sklearn import preprocessing
6. %matplotlib inline
7. import matplotlib.pyplot as plt
```

Kode Program 1 merupakan pemanggilan *library* yang akan digunakan untuk membaca data sampai pada operasi matematika dengan python nantinya. *Library* yang digunakan diantaranya *pandas*, *pylab*, *numpy*, *scipy.optimize*, *preprocessing* dan *matplotlib*. Selanjutnya pada Kode Program 2 akan dilakukan pembacaan file XYZ\_Kupang.csv. File yang akan dibaca kemudian ditampilkan 5 baris pertamanya yang dimulai dari index ke-0 sampai index ke-4.

### Kode Program 2. Membaca File .csv

```
1. churn_df = pd.read_csv("XYZ_Kupang.csv")
2. churn_df.head()
```

Hasil dari Kode Program 2 kemudian dapat dilihat pada Gambar dimana semua *field* data transaksi ditampilkan. *Field* data dimulai dari MID\_INDUK sampai pada MCC\_VISA yang mana hampir semua *field* dalam bentuk numerik sesuai syarat *Logistic Regression* yang akan digunakan untuk pengolahan. Pada Gambar terlihat juga data kategorik sesuai syarat dari *Logistic Regression* pada kolom NOP yaitu 1 dan 2. Selanjutnya dari data transaksi yang sudah ditampilkan pada Gambar 3 ini, akan dipilih lagi beberapa *field* yang datanya menggunakan numerik agar dapat diproses lebih lanjut.

	MID_INDUK	MID	TID	SN_EDC	ALAMAT	VOLUME	PRODUKTIF_TRANSAKSI	TIDAK PRODUKTIF_TRANSAKSI	NOP	MDR_ON_US	MDR_OFF_US	MDR DEBIT	MCC MASTER	MCC VISA
0	1.986900e+09	1986900023	10132991	324-022-196	JL. SUVERDI RT. 028 / RW. 006,	50000000	21	21	1	0.8	1.6	00,15	4215	4215
1	1.986900e+09	1986900023	10132992	324-028-097	JL. SUVERDI RT. 028 / RW. 006,	0	0	0	1	0.8	1.6	00,15	4215	4215
2	1.986900e+09	1986900023	10132993	324-028-102	JL. SUVERDI RT. 028 / RW. 006,	25000000	11	11	1	0.8	1.6	00,15	4215	4215
3	1.986900e+09	1986900023	10132994	324-022-192	JL. SUVERDI RT. 028 / RW. 006,	10000000	20	20	1	0.8	1.6	00,15	4215	4215
4	1.999028e+09	1999028144	10158401	324-026-990	KEL. NAIONAT KUPANG TIMUR	150000000	100	100	2	1.8	2.0	00,15	5983	5983

**Gambar 3.** Data .csv yang berhasil dibaca

Untuk melakukan pemilihan *field* data transaksi pada Bank XYZ maka perlu dilakukan pemilihan beberapa *field* yang hanya dalam bentuk numerik nilainya. Kode Program 3 merupakan perintah untuk memilih *field* yang isinya adalah numerik diantaranya TID, VOLUME, PRODUKTIF\_TRANSAKSI, MDR\_ON\_US MDR\_OFF\_US dan NOP. Dalam hal ini NOP sebagai variabel Dependen dan *field* lainnya adalah variabel independen. Terlihat pada *code* 3 juga dimana *field* NOP diset untuk tipe integer agar dapat dilakukan *modeling* nantinya.

#### Kode Program 3. Pemilihan *field* numerik

```

1. churn_df = churn_df[['TID', 'VOLUME', 'PRODUKTIF_TRANSAKSI', 'MDR_ON_US',
'MDR_OFF_US', 'NOP']]
2. churn_df['NOP'] = churn_df['NOP'].astype('int')
3. churn_df.head()
```

Hasil dari Kode Program 3 dapat dilihat pada Gambar 4 di bawah ini, dimana *field* yang terpilih saja yang ditampilkan untuk nantinya dilakukan *modeling*. Terlihat semua data ditampilkan dalam bentuk numerik yang mana dapat diproses untuk *modeling* dengan *Logistic Regression*.





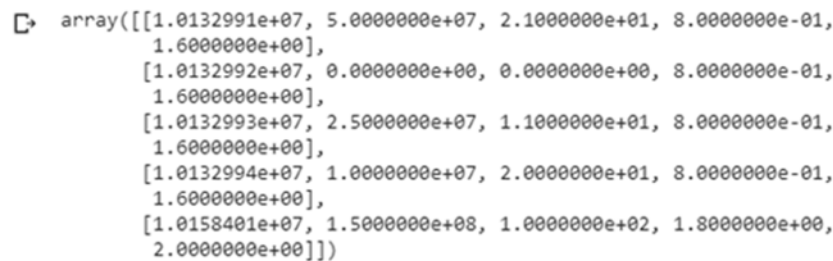
	TID	VOLUME	PRODUKTIF_TRANSAKSI	MDR_ON_US	MDR_OFF_US	NOP
0	10132991	50000000	21	0.8	1.6	1
1	10132992	0	0	0.8	1.6	1
2	10132993	25000000	11	0.8	1.6	1
3	10132994	10000000	20	0.8	1.6	1
4	10158401	150000000	100	1.8	2.0	2

**Gambar 4.** Hasil seleksi beberapa *field* yang akan dianalisa

Untuk kepentingan modeling maka *field* yang sudah terpilih perlu ditampilkan dalam bentuk *array*. Kode Program 4 merupakan perintah untuk menampilkan array menggunakan *library* numpy untuk operasinya. *Array* yang ditampilkan hanya berjumlah 5 buah saja. Hasil dari Kode Program 4 ini dapat dilihat pada Gambar 5 di bawah ini dimana array hanya 5 baris teratas saja yang ditampilkan namun semua data akan diproses untuk modeling dari setiap variabel independennya.

**Kode Program 4.** Menampilkan dalam bentuk *array*

```
1. X = np.asarray(churn_df[['TID', 'VOLUME', 'PRODUKTIF_TRANSAKSI', 'MDR_ON_US',
    'MDR_OFF_US']])
2. X[0:5]
```



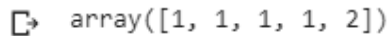
```
array([[1.0132991e+07, 5.0000000e+07, 2.1000000e+01, 8.0000000e-01,
    1.6000000e+00],
    [1.0132992e+07, 0.0000000e+00, 0.0000000e+00, 8.0000000e-01,
    1.6000000e+00],
    [1.0132993e+07, 2.5000000e+07, 1.1000000e+01, 8.0000000e-01,
    1.6000000e+00],
    [1.0132994e+07, 1.0000000e+07, 2.0000000e+01, 8.0000000e-01,
    1.6000000e+00],
    [1.0158401e+07, 1.5000000e+08, 1.0000000e+02, 1.8000000e+00,
    2.0000000e+00]])
```

**Gambar 5.** Hasil dalam bentuk *array*

Begitu juga dengan variabel dependennya (NOP) dimana perlu untuk diubah dulu ke dalam *array* dan ditampilkan baris atasnya. Jumlah indexnya adalah 5 namun semua data *field* ini nantinya akan tetap diproses untuk *modeling*. Hasil dari tampilan *array* dapat juga dilihat pada Gambar 6.

**Kode Program 5.** Mengubah ke *array*

```
1. y = np.asarray(churn_df['NOP'])
2. y [0:5]
```



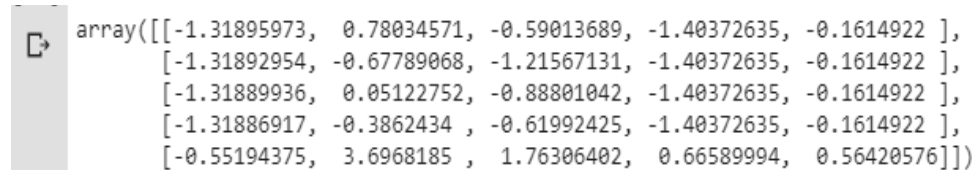
```
array([1, 1, 1, 1, 2])
```

**Gambar 6.** Hasil dalam bentuk *array*

Selanjutnya akan dilakukan pemodelan dengan melakukan *preprocessing* terlebih dahulu. Kode Program 6 merupakan perintah untuk melakukan pemodelan dengan menggunakan *library* sklearn dan nantinya akan dilakukan *preprocessing*. Fungsi dari *preprocessing* ini adalah melakukan normalisasi pada dataset yang ada agar nantinya dapat dilakukan *modeling* dengan *Logistic Regression*. Hasil dari normalisasi dataset ini dapat dilihat pada Gambar 7 di bawah ini dimana ditampilkan dalam bentuk *array* beberapa baris saja.

**Kode Program 6.** Memulai pemodelan *preprocessing*

```
1. from sklearn import preprocessing
2. X = preprocessing.StandardScaler().fit(X).transform(X)
3. X[0:5]
```



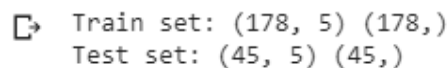
```
array([[ -1.31895973,  0.78034571, -0.59013689, -1.40372635, -0.1614922 ],
       [ -1.31892954, -0.67789068, -1.21567131, -1.40372635, -0.1614922 ],
       [ -1.31889936,  0.05122752, -0.88801042, -1.40372635, -0.1614922 ],
       [ -1.31886917, -0.3862434 , -0.61992425, -1.40372635, -0.1614922 ],
       [ -0.55194375,  3.6968185 ,  1.76306402,  0.66589994,  0.56420576]])
```

**Gambar 7.** Hasil *Preprocessing*

Setelah melakukan *preprocessing* maka perlu dilakukan tahapan selanjutnya yaitu train dataset untuk melakukan pelatihan terhadap dataset yang ada dan *test* dataset untuk melakukan tes pada dataset yang ada. Jumlah train dataset adalah 80% dan *test* dataset adalah 20%. Pemanggilan *library* bisa dilihat pada Kode Program 7 sedangkan hasil dari train dataset dan *test* dataset dapat dilihat pada Gambar 8 di bawah ini.

**Kode Program 7.** Train dataset

```
1. from sklearn.model_selection import train_test_split
2. X_train, X_test, y_train, y_test = train_test_split( X, y, test_size=0.2,
    random_state=4)
3. print ('Train set:', X_train.shape,  y_train.shape)
4. print ('Test set:', X_test.shape,  y_test.shape)
```



```
Train set: (178, 5) (178,)
Test set: (45, 5) (45,)
```

**Gambar 8.** Hasil *train* dan *test* dataset.

Tahapan selanjutnya adalah melakukan *modeling* dengan menggunakan Scikit-learn dan pendekatan *Logistic Regression*. Kelebihan dari *Logistic*

*Regression* dengan *library* ini adalah kemampuannya menyelesaikan permasalahan *overfitting* dalam *machine learning*. *Modeling* dengan *Logistic Regression* dapat dilihat pada *Code 8* dimana ada parameter *C* yang bernilai 0.01. Maksud nilai parameter *C* ini adalah untuk menunjuk invers dari kekuatan regulasi yang mana harus bernilai *float* positif yaitu 0.01. Nilai yang lebih kecil menentukan regularisasi yang lebih kuat dari pemodelan yang akan dilakukan. Hasil dari *code 8* (pemanggilan *library*) dapat dilihat pada Gambar 9 di bawah ini.

#### Kode Program 8.

```
1. from sklearn.linear_model import LogisticRegression
2. from sklearn.metrics import confusion_matrix
3. LR = LogisticRegression(C=0.01, solver='liblinear').fit(X_train,y_train)
4. LR

↳ LogisticRegression(C=0.01, class_weight=None, dual=False, fit_intercept=True,
    intercept_scaling=1, l1_ratio=None, max_iter=100,
    multi_class='warn', n_jobs=None, penalty='l2',
    random_state=None, solver='liblinear', tol=0.0001, verbose=0,
    warm_start=False)
```

**Gambar 9.** Pemodelan.

Selanjutnya pada Kode Program 9 dilakukan prediksi data set yang sudah disiapkan untuk tes. *Test set* ini ditampilkan dalam bentuk array yang mana nilainya dalam bentuk kategori yaitu 1 dan 2 (data NOP). Hasil dari *code 8* dapat dilihat pada Gambar di bawah ini.

#### Kode Program 9. Prediksi *dataset*

```
1. yhat = LR.predict(X_test)
2. yhat

↳ array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1,
    2, 1, 1, 1, 2, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1,
    1])
```

**Gambar 10.** Hasil *array* NOP.

Tahapan berikutnya adalah melakukan prediksi probabilitas seperti yang ditunjukkan oleh Kode Program 10. Data *test* NOP selanjutnya dilihat probabilitasnya dan hasilnya seperti pada Gambar 11, dibagi menjadi dua kolom. Kolom pertama adalah probabilitas dari kelas  $P(Y=1|X)$  atau sesuai data NOP bernilai 2 sedangkan kolom kedua adalah probabilitas dari kelas  $P(Y=0|X)$  atau sesuai data NOP bernilai 1.

#### Kode Program 10. Prediksi Probabilitas

```
1. yhat_prob = LR.predict_proba(X_test)
2. yhat_prob
```

```
array([[0.57433966, 0.42566034],  
       [0.64670169, 0.35329831],  
       [0.68238111, 0.31761889],  
       [0.67162752, 0.32837248],  
       [0.58707013, 0.41292987],  
       [0.63325502, 0.36674498],  
       [0.56810951, 0.43189049],  
       [0.6781368 , 0.3218632 ],  
       [0.51329985, 0.48670015],  
       [0.57439864, 0.42560136],  
       [0.59297013, 0.40702987],  
       [0.67192361, 0.32807639],  
       [0.45928962, 0.54071038],  
       [0.57731713, 0.42268287],  
       [0.57640575, 0.42359425],  
       [0.68128419, 0.31871581],  
       [0.51866022, 0.48133978],  
       [0.70114211, 0.29885789],  
       [0.5129541 , 0.4870459 ],  
       [0.66351781, 0.33648219],  
       [0.63850992, 0.36149008],  
       [0.64906866, 0.35093134],  
       [0.48258711, 0.51741289],  
       [0.64804257, 0.35195743],  
       [0.56751528, 0.43248472],  
       [0.62883718, 0.37116282],
```

**Gambar 11.** Hasil probabilitas NOP.

Untuk melihat evaluasi dari akurasi hasil *test* di atas maka selanjutnya pada Kode Program 11 dilakukan pengolahan *jaccard index*. Dari dua kolom pada hasil sebelumnya, nilai akurasi akan bernilai sempurna apabila mendekati 1 sebaliknya tingkat akurat rendah apabila mendekati 0. Hasil dari Kode Program 11 dapat dilihat pada Gambar 12 , dimana hasil akurasi menunjukkan nilai 0.88 yang artinya tinggi karena mendekati 1.

**Kode Program 11.** Akurasi *jaccard index*

```
1. from sklearn.metrics import jaccard_similarity_score  
2. jaccard_similarity_score(y_test, yhat)
```

```
/usr/local/lib/python3.6/dist-packages/sklearn/metrics/classification.py:635: DeprecationWarning:  
'and multiclass classification tasks.', DeprecationWarning)  
0.8888888888888888
```

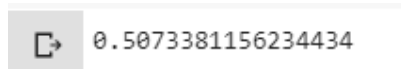
**Gambar 12.** Hasil akurasi.

Terakhir, dilakukan pencarian nilai *log loss* untuk evaluasi. Rentan nilai untuk evaluasi ini adalah antara 0 sampai 1 yang mana 1 adalah kesempurnaan dari

evaluasi. Dari *test* yang dilakukan dengan pendekatan *Logistic Regression* untuk memprediksi NOP yang terjadi pada Bank XYZ dilakukan dengan perintah pada Kode Program 11. Hasilnya tampak pada Gambar 13 di bawah dimana nilai yang dihasilkan adalah 0,507 yang artinya lebih dari setengah untuk evaluasinya NOP yang terjadi pada Bank XYZ. Sehingga dari hasil ini berarti bahwa mesin ATM yang memiliki nilai prediksi tinggi perlu untuk dipindah agar tidak membuang sumber daya. Hal ini tentu akan membantu pihak Bank dalam merencanakan dengan baik pemasangan mesin ATM di lokasi tertentu secara lebih tepat sasaran.

**Code 12.** Nilai *log loss* untuk evaluasi.

```
1. from sklearn.metrics import log_loss
2. log_loss(y_test, yhat_prob)
```

0.5073381156234434

**Gambar 13.** Hasil prediksi

## Simpulan

Dari penelitian ini dibuktikan bahwa prediksi terhadap NOP pada mesin ATM yang terjadi di Bank XYZ dapat diprediksi sehingga pihak Bank dapat menentukan lokasi penempatan mesin ATM tersebut. Mesin ATM yang ditempatkan kemudian dapat menjadi tepat sasaran dalam melayani kebutuhan transaksi *non-tunai* di *merchant* tertentu. Pendekatan *Machine Learning* dalam menyelesaikan persoalan penelitian ini dapat tercapai dengan menggunakan *Logistic Regression* sebagai salah satu algoritma *Supervised Learning*. NOP pada Bank XYZ dapat diprediksi dengan tingkat kemungkinan 0.507. Hal ini membuktikan bahwa NOP yang terjadi pada Bank XYZ dapat diprediksi dengan tingkat kemungkinan yang melebihi 50%. Ke depannya apabila ada data nasabah atau *non-nasabah* yang melakukan transaksi di mesin ATM Bank XYZ, maka akan dengan mudah dilakukan prediksi bagaimana NOP yang akan terjadi pada mesin-mesin ATM. Adapun saran penelitian ke depan yaitu dapat menggunakan pendekatan *Machine Learning* dengan metode yang berbeda seperti *Time Series*. Hal ini dilakukan agar dapat menganalisis dan memprediksi data dengan tipe data runtutan waktu dari bulan ke bulan dan tahun ke tahun.

## Daftar Pustaka

- [1] <https://ekonomi.kompas.com/read/2017/08/31/143104926/presiden-jokowi-soroti-jumlah-mesin-atm-yang-terlalu-banyak> diakses tanggal 4 Mei 2019.
- [2] <https://www.investopedia.com/terms/m/merchant-discount-rate.asp> diakses tanggal 4 Mei 2019.

- 
- [3] G.M. Fitzmaurice, N.M. Laird., 2001. Multivariate Analysis: Discrete Variables (Logistic Regression), International Encyclopedia of the Social & Behavioral Sciences.
  - [4] <https://medium.com/@makersinstitute/pengenalan-terhadap-machine-learning-9011fe71d1e4> diakses tanggal 4 Mei 2019.
  - [5] <https://www.statistikian.com/2015/02/regresi-logistik.html> diakses tanggal 4 Mei 2019.
  - [6] Hunt J. 2019. Advanced Guide to Python 3 Programming, Springer.
  - [7] <https://www.python.org/about/> diakses tanggal 19 Desember 2019.
  - [8] Le. H.H, Viviani J.L. 2018. Predicting bank failure: An improvement by implementing a machine-learning approach to classical financial ratios. Volume 44, April 2018, Pages 16-25. Research in International Business and Finance, Elsevier.
  - [9] Ginting. W.A, dkk, 2019. Faktor-Faktor Pelaporan Keuangan Berbasis Website Dengan Model Regresi Logistik. JURNAL AKUNTANSI DAN BISNIS: Jurnal Program Studi Akuntansi. ISSN 2503-0337 (*Online*)
  - [10] Kasiram, Moh. 2008. Metodologi Penelitian. Malang: UIN-Malang Pers.
  - [11] Sugiyono. 2012. Metode Penelitian Kuantitatif Kualitatif dan R&D. Bandung: Alfabeta. Prof.H. Rozali Abdulah, S.H.2005.